VZ 2000 NEWSLETTER

Laser 110 - Laser 200 - Laser 210 - Laser 310 - VZ200 - VZ300 - Texet TX8000 - Salora Fellow

Issue #1 January 2000

By Guy Thomason

Introduction

Welcome to the first edition of the VZ 2000 newsletter! The VZ series of computers has recently undergone a second lease of life, thanks to the world wide reach of the Internet. A few years ago (late 1995) I got connected to the net. The concept of doing topic searches by entering a few key words was marvellous, kind of like a gigantic online encyclopedia.

I was still interested in the VZ back then, so it wasn't long before I started doing searches on "VZ200". At that time, not surprisingly, I didn't get any results and kind of gave up. I figured the computer was too old and obscure for anyone else to really care about it anymore.

By around 1997 computer hardware was starting to get more powerful, and a fellow from work came in one day and showed us a program that turned his computer into a commodore 64. He was pretty excited about this, because he said he used to love some of the games on it, but it was too much trouble unpacking it and all the assorted cables, disks, joysticks and other stuff from his closet where it lived for years collecting dust. Another guy at work used to own an Atari 800, so he went away and did some searches and the next day he came in all excited and said he found an emulator for that one too. Then someone else got one for the apple II.

This started to intrigue me and I began doing searches on "emulators". Before too long I stumbled onto something called MAME, which would play all the old arcade games! Now this was just mind blowing. As a youngster I got hooked on video games since my first game of space invaders, and over the years must have put a fortune in coins into the slots of these creations. And now I could play them on my computer! I fired up GALAGA, and....whoa! Isn't that slow! Now I had justification for my next upgrade, a pentium motherboard and 233 mhz chip.

At about the same time I was still occasionally searching for anything on the VZ200, and one day found Brian Murray's VZ page. I also found an emulator for the MSX computer, written by a clever person named Marat Fayzullin. The biggest gem about the MSX emulator was that the source code was also included! So now I had a program which would emulate the Z80, and an example of how to put it together to emulate an entire computer system. I figured since no one else had written one for the VZ, I would try to write one based on Marat's MSX emulator.

I got in touch with Brian who had managed to extract the VZ Rom, and started coding. I had also saved up enough for the pentium motherboard and chip, so one weekend I lashed out and bought it home, and excitedly installed it into my computer case. As we all know, manuals are rather unnecessary items that manufacturers throw in to clutter things up. In my haste to get my new pentium powered PC running, I plugged the power cable into the motherboard the wrong way, and when I turned on my computer I was treated to a bright green flash, a very loud bang and that nice expensive smell of fried components.

No more computer. No more harddrive. No more Galaga. No more VZ project. No backups.

Some months later I managed to rebuild my PC. I still wanted to get a VZ emulator going, and I started doing searches again. I found some more VZ pages, including George Lui's site, which really impressed me because of all the work that had gone into it, and the amount of information and pictures there. I was very interested in the VZ again now, especially since I knew other people were still interested in it as well. There was also a list of email addresses floating around of several other people that were still talking about the VZ. I sent out an email to these people talking about my intent to write a VZ emulator, and got the V2.0 Rom from Brian again, as well as from Dave Maunder.

After a bit more coding I got to the stage where the emulator would start, load in the rom and display the startup message. This was originally in vga dos text mode, ie just using the vga character set. I can't begin to explain how rewarding this was, though. The emulator didn't really do anything yet - wouldn't read the keyboard, make beeps or anything, it just displayed:

```
VIDEO TECHNOLOGY
BASIC V2.0

READY
```

...But this was such a big deal to me I couldnt sleep.

Enter The Dragon, July 1999

In my searches I had also found an emulator for the Dragon 32/CoCo Computer. This was significant because these machines used the same display chip as the VZ, namely the Motorolla 6847. I managed to get a screen dump of this emulator running, and extracted from the bitmap the character set for the VZ. After writing some code for the VZ keyboard handling, and adding the character set, I finally had a clunky, but working, emulator for the VZ.

In the 6 months since I first got the emulator working, nearly all of the DSE software library for the VZ that was available in Australia has been converted to run on the emulator. A new VZ driver for the Mess emulator has been written by Juergen Buchmueller, who has also added disk support for the VZ.

...But why would you want to?

This is a common question, and it is always asked from people whose first computer was an IBM clone. There is a special kind of appeal with these old 1980s micros, that is hard to explain unless you owned one. I like emulators because:

- It is a fun and interesting hobby
- They preserve an important part of computing history, that would otherwise be lost when the original hardware eventually fails
- They allow old computer software & games to be run again, on your existing computer, without having to set up the old computer. (My notebook computer functions as a PC, Amiga, Vectrex and VZ300, and fits in my briefcase.)
- In many cases emulators do a better job than the original computer. The VZ for instance didnt give a very clear tv picture, displayed static on the screen whenever there was graphics moving around, and sometimes gave a LOADING ERROR when trying to load a program from cassette

So What Next?

I'm still as enthusiastic about the VZ today as I was when I got my '200 in 1984. I intend to continue developing the emulator and writing VZ software for at least another 16 years.

This Issues Feature - The VZ Tape Format

Prior to 1985 the VZ floppy drive was not available in Australia and all File I/O was via the cassette tape interface. This article will discuss how the VZ saves data to tape.

VZ Rom routines

The VZ Rom has a tape save routine located at address 34A9h. The routine is coded to save Basic programs, however with a small change to the way the routine is called it is easy to save machine code programs or custom data files. (See note 1)

Data written to tape is saved in the following order:

Synch bytes
Preamble
Filetype
Filename
Start Address
End Address
Program bytes
Checksum
Trailer

Synch bytes

The synch bytes are written as 255 bytes of 80h. (10000000b) The synch bytes signal the start bit of the tape data stream, and help to buffer the signal noise which occurs at the start of each recording.

Preamble

5 bytes of FEh are written to tape following the synch bytes. During loading, The preamble bytes confirm the tape is synched correctly, and the tape being read is VZ format.

Filetype

Programs are saved with a default filetype of F0h. The VZ rom uses F0h to identify Basic programs. The filetype also tells the interpreter what to do on completion of loading - return control to the user for basic programs, or auto execute for m/c programs (F1h)

Filename

The filename is written next - up to 17 bytes, terminated by 00h. The filename is taken from a buffer at address 7A9Dh

Start Address

The start address is read from the values at address 78A4/5. For basic programs the start address is normally 7AE9h

End address

The end address is read from the values at address 78F9/A.

Program bytes

all bytes between the start and end address inclusive are written as the main data block.

Checksum

During the save operation each program byte is added to a 16 bit checksum at address 7823/4h. This is compared with the checksum generated in verify & load operations to check the program was reloaded correctly.

Trailer

A final 20 bytes of 00h is written after the checksum. This is again used to buffer the sharp noise caused by the loss of signal, which could potentially corrupt the checksum.

Note 1: saving other program types

The start of the save listing is shown below.

```
34A9: DI
34AA: LD C,F0h
34AC: CALL 3558h
```

The first 3 instructions disable interrupts to allow for perfect timing, load the C register with the filetype (F0h for basic program), and call the routine at address 3558h which writes the synch bytes, preamble, filetype, and filename. To save a machine code program, or any custom data file:

1. Load the start and end address into 78A4/5 & 78F9/A
2. Move the filename (null terminated) into the buffer at 7A9D
3. Disable interrupts
4. Load the C register with your filetype
5. Call 34AC

Loading Programs

To load a program, the VZ does the following:

1. Scans the tape stream for the first synch byte of 80h
2. Waits for first preamble byte
3. Reads 4 more preamble bytes, (if these aren't found a loading error is generated)
4. Reads filetype
5. Reads filename
6. Reads Start Address
7. Reads End Address
8. Reads Each program byte into memory, and adds byte to checksum
9. Reads Checksum
10. Compares tape checksum to generated checksum, generates a loading error if not equal
11. Checks filetype, if F1 jumps to m/c start address

Code Corner

This section of the newsletter gives helpful program routines that were contributed by authors. For this edition, there are 2 pixel plotting routines. The first is by Juergen Buchmueller, and the second is by Guy Thomason. These routines are virtually identical. (There's only so many ways to draw a dot!)

Contributions to this section are always welcome, if you have written a nifty bit of code lately, send it to me at intertek@powerup.com.au and I will steal it and take credit for it. (Just kidding)

Example 1

```
; plot pixel at x=L, y=H, color=C
plot:
push af
push bc
push de
push hl

plot0:
ld a, h
and 63
ld h, a

ld a, l
and 127
ld l, a

ld a, l
; calculate screen offset
; tricky, eh? ;)
sll l
srl h
rr l
srl h
rr l
srl h
rr l

; now add the screen memory base
ld de, (sprmem)
add hl, de

and 3 ; pixel offset
inc a
ld b, 11111100b
plot1:
rrc b
rrc b
rrc c
rrc c
dec a
jr nz, plot1
ld a, (hl)
and b
or c
ld (hl), a
plotx:
pop hl
pop de
pop bc
pop af
ret
```

Example 2

```
void pset(char x, char y, char c)

{

asm{

        LD L, (IX-02)           ; get Y coordinate

        LD H,0                  ; make 16 bit

        ADD HL,HL               ; multiple * 2

        ADD HL,HL               ; multiple * 4

        ADD HL,HL               ; multiple * 8

        ADD HL,HL               ; multiple * 16

        ADD HL,HL               ; multiple * 32

        LD DE,28672             ; start of screen mem

        ADD HL,DE               ; DE has address of screen row
```

```
        LD A,(IX-03)            ; get X coordinate
        LD E,A                  ; setup for 16 bit add
        SRL E                   ; divide by 2
        SRL E                   ; divide by 4
        LD D,0
        ADD HL,DE               ; DE points to correct screen byte
        SLA E                   ; multiply by 2
        SLA E                   ; multiply by 4
        SUB E                   ; A has remainder - 0,1,2,3
        LD B,A                  ; set up for djnz
        INC B
        LD C,(IX-01)            ; get color
        LD D,11111100b          ; pixel mask
PSET1:  RRC C
        RRC C                    ; adjust color
        RRC D
        RRC D                   ; adjust pixel mask
        DJNZ PSET1              ; C has color, D has mask
        LD A,(HL)               ; get byte
        AND D                   ; mask out pixel
        OR C                    ; set pixel to color
        LD (HL),A               ; and display
};
}
```

Software Archive

Listed below is the current list of commercial software titles that have been converted to PC files for use in the emulators.

Hamburger Sam

Dracula

Digout

Invaders

Hoppy

Ghost Hunter

Missile Attack

Lunar Lander

Galaxon

Penguin

Circus

Duel

Space Ram

Planet Patrol

Tennis

Kamikazi

Dawn Patrol

Knights & Dragons

VZ Panik

Meteor

Super Snake

Ladder Challenge

Star Blaster

Chess

Defence Penetrator

VZ Contacts

Australia

Guy Thomason - intertek@powerup.com.au
Jason Oakley - illogix@ihug.com.au

Germany

Juergen Buchmueller - pullmoll@t-online.de

If you would like to be listed as a contact please give me your permission. Being a "contact" basically means you don't mind people asking you technical type VZ questions.